LA-UR-21-29163

Title: A novel MC TRT method: vectorizable variance reduction for the energy spectra

Author(s): Morgan, Jackson Porter
Long, Alex Roberts
Long, Kendra Phyllis

Intended for: Student's records / future presentations internal to university

Issued: 2021-09-16

A novel MC TRT method: vectorizable variance reduction for the energy spectra

Jackson P. Morgan
Mentors: Alex Long and Kendra Long

September 9th, 2021

Contents

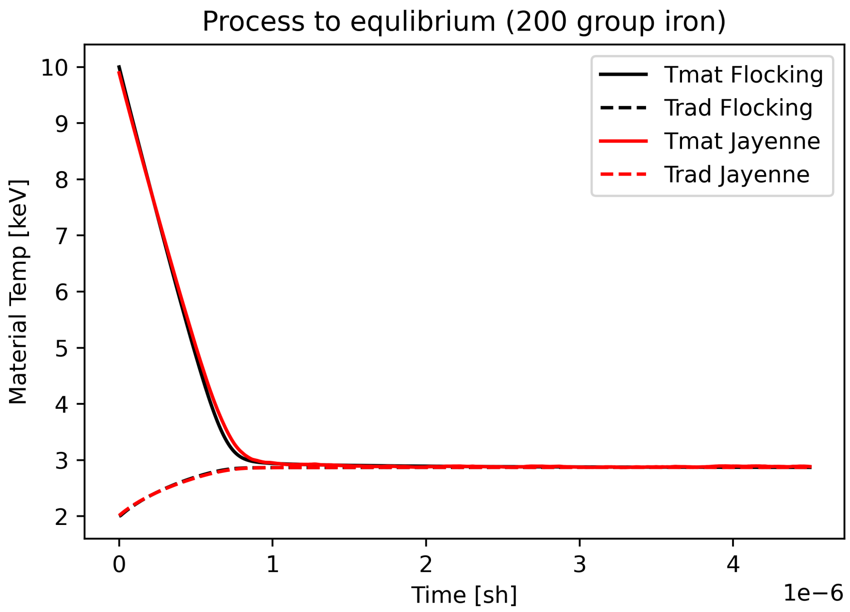
Introduction

- TRT Equations solved via Monte Carlo Methods
- Jayenne implements multigroup IMC and can have a hard time resolving energy space
- Newer machines have more vector lanes
- This work is preliminary and will be continued over the next few months

Method Description

- Qualitative description
- TRT justification in back matter
- flow chart of *flocking* test bed

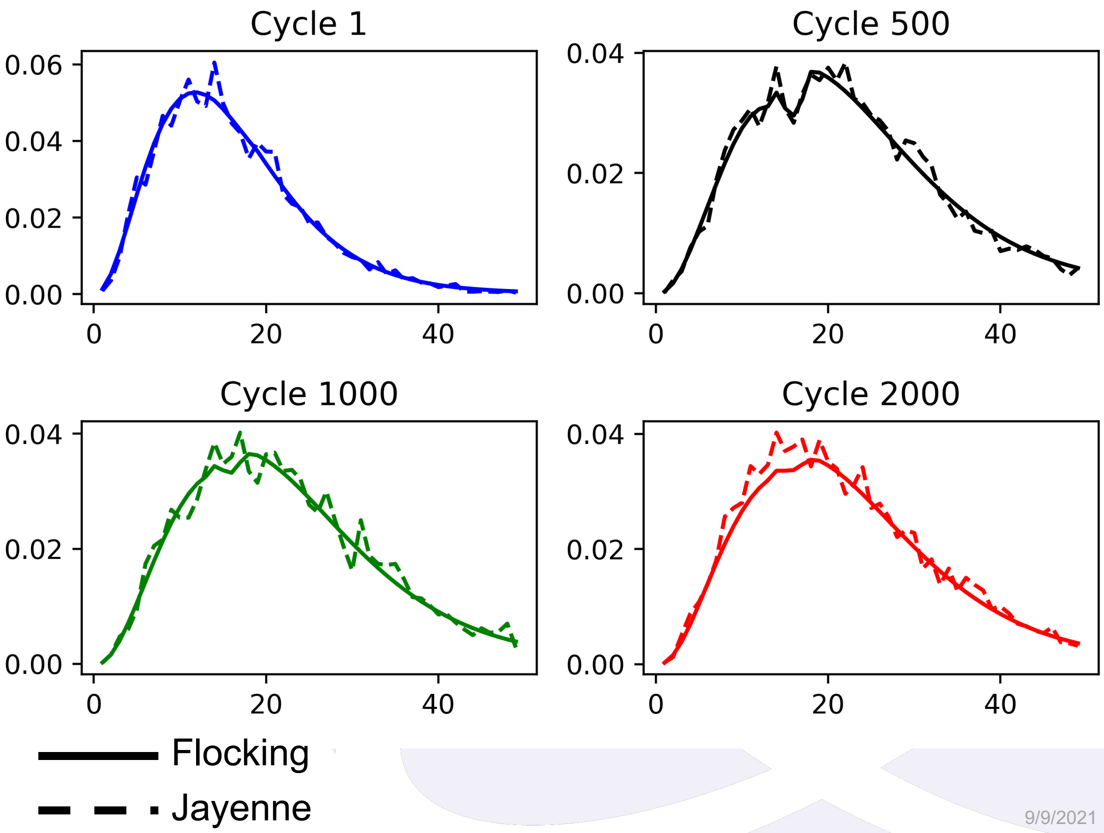
Method Confirmation (Testcase: *Toasty*)



- Heating profile between either methods match very closely
- $T_{RAD,0}$ = 2.0 keV
- $T_{MAT,0}$ = 10.0 keV
- C_v = 0.1 jk/g-keV
- ρ = 1 g/cm^3

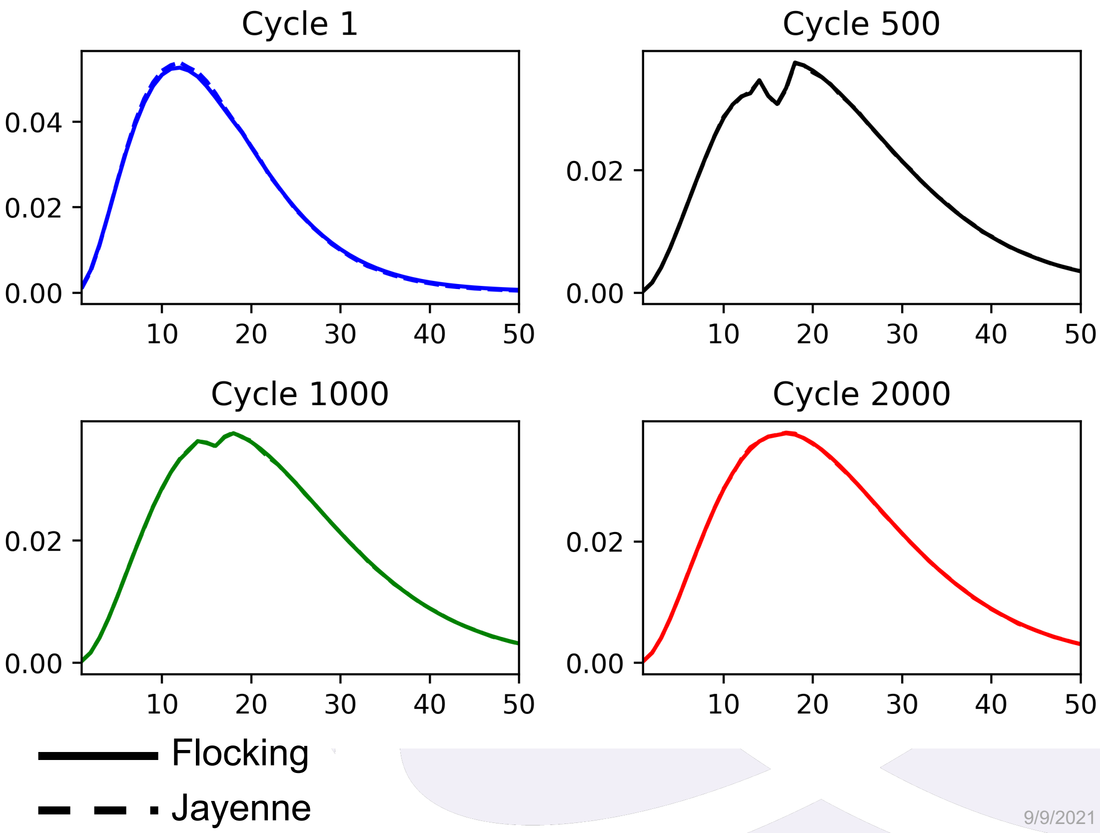
Spectra Comparison (Testcase: *Toasty*)

- Flocking: 10k particles
  Jayenne: 10k particles
- 200 iron group material data
- $T_{RAD,0}$ = 2.0 keV
- $T_{MAT,0}$ = 10.0 keV



Spectra Comparison (Testcase: *Toasty*)

- Flocking: 10k particles
  Jayenne: 5 million particles
- 200 iron group material data
- $T_{RAD,0} = 2.0$ keV
- $T_{MAT,0} = 10.0$ keV



Code Sample Parallelizable Loop

- OpenMP SIMD based pragmas
- 9 pragma statements
- Vectorization reports where leveraged to see how well auto-vectorization worked
- Code Sample:

```
#pragma omp simd
for(int g=0; g<num_groups; g++)
{
    double ew_factor =          exp(-interp_mgp[g] * d_event);
    double ewpl_factor =        (1.0 - ew_factor) / (interp_mgp[g] * d_event);
    abs_E +=                    all_particles[i].group_ew[g] * (1.0 - ew_factor);
    rad_E_tracks +=             all_particles[i].group_ew[g] * d_event * ewpl_factor;
    all_particles[i].group_ew[g] *=   ew_factor;
}
```

Runtimes with Various Compiler Flags: Broadwell

- 60% performance increase from no vectors (1) to full vectorization and fast (5) (intel/18.0.3)
- 64% performance increase from no vectors (1) to full vectorization and fast (5) (dpcpp)
- Shows that vectorization can yield performance boons

	Compilation Flags	Intel/18.0.3 (icpc)	Intel/2021.3.0 (dpcpp)
1	-O2 –no-vec	188.68	193.08
2	-O2	119.22	168.60
3	-fast	108.75	78.64
4	-O2 -fopenmp	99.58	96.23
5	-fast -fopenmp	74.95	69.38

Runtime Vectors on/off Skylake Gold

- This shows again increased performance when vectors are turned on and fast is allowed to let loose (64% performance increase)
- We did expect more performance increases due to AVX512

Compiler Condition	Runtime [s]
-O2 –no-vec	153.56
-O2 -fopenmp	78.81
-fast -fopenmp	55.60

Runtime Comparisons to Jayenne

- Same number of particle, timesteps, groups, Broadwell node. Using full vectorization for flock
- Jayenne takes ~20X more particles to get to the same energy spectra resolution

Code	Toasty	Cool
Jayenne	7.64	22.37
Flocking	16.33 (+53%)	69.38 (+68%)

Conclusions

- Further work is required to fully flesh out whether or not this method is fully viable given current production codes
- Data produced up to now gives us confidence that we will continue to see variance reduction behavior at amenable runtimes as we move this test bed forward
- This method has characteristics of event and history based Monte Carlo transport, without the bulky modifications required by event based Monte Carlo while avoiding some overhead

Future Work

- Produce a figure of merit
- Move into a 1 spatial dimensional space
- Cluster groups by thickness (avoid forcing collisions)
- Test with Marshak wave

Acknowledgements

Special thanks to my LANL Mentors:

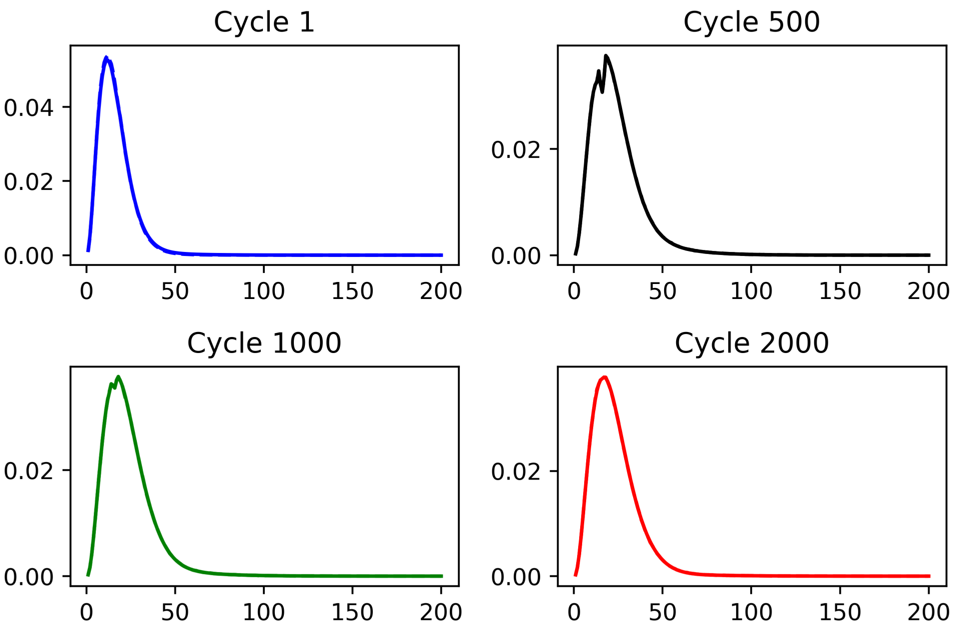
- Alex Long
- Kendra Long
- Simon Bolding

Center for Exa-scale transient Monte-carlo Neutron Transport (CEMeNT)

PSAAP-III Program

Full Group Plot for *Toasty* Testcase

- Flocking: 10k particles
  Jayenne: 5 million particles
- 200 iron group material data
- $T_{RAD,0} = 2.0$ keV
- $T_{MAT,0} = 10.0$ keV



TRT Equation Justification 1/2

Start w/ TRT:

$$\frac{1}{c}\frac{\delta I_g(t)}{\delta t} + \sigma_{a,g} I_g(t) + \sigma_{s,g} I_g(t) = \chi_g(T) B_g(T)$$

Define a 'time-continuous' gray scattering opacity:

$$\bar{\sigma}_s(t) = \frac{\sum \sigma_{s,g} I_g(t)}{\sum I_g(t)}$$

"Multiply by 1":

$$\frac{1}{c}\frac{\delta I_g(t)}{\delta t} + \sigma_{a,g} I_g(t) + \frac{\bar{\sigma}_s(t)}{\bar{\sigma}_s(t)} \sigma_{s,g} I_g(t) = \chi_g(T) B_g(T)$$

Regroup terms for clarity:

$$\frac{1}{c}\frac{\delta I_g(t)}{\delta t} + \sigma_{a,g} I_g(t) + \bar{\sigma}_s(t) \left[\frac{\sigma_{s,g}}{\bar{\sigma}_s(t)} I_g(t)\right] = \chi_g(T) B_g(T)$$

Los Alamos NATIONAL LABORATORY

TRT Equation Justification 2/2

Assuming I sample distance-to-scatter with the gray value, I need to adjust group-wise energy-weights by:

$$ew_{1,g} = \left[\frac{\sigma_{s,g}}{\bar{\sigma}_s(t)}\right] ew_{0,g}$$

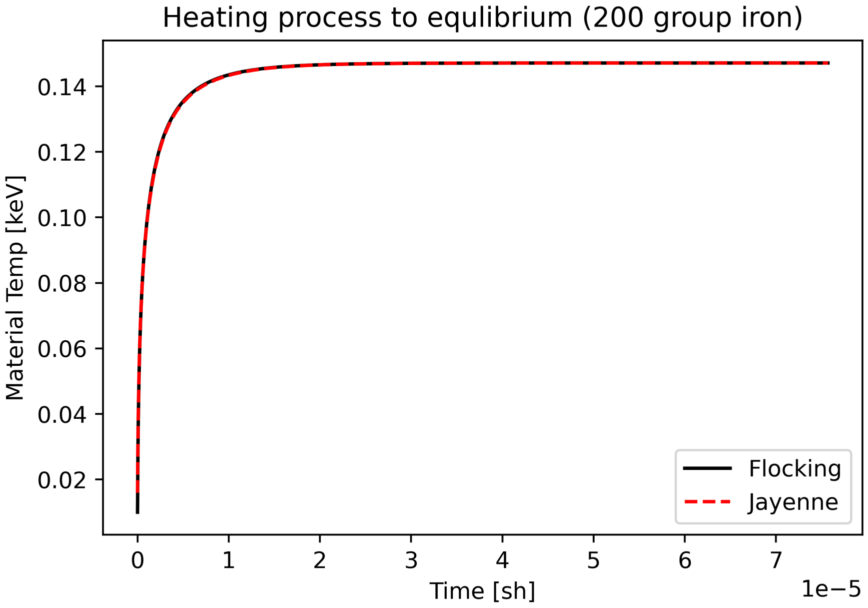
This conserves total energy-weight. To show, plug in orig. expression for gray scattering opacity

$$ew_{1,g} = \left[\frac{\sigma_{s,g} \sum ew_{0,g'}}{\sum \sigma_{s,g'} ew_{0,g'}}\right] ew_{0,g} \quad \rightarrow \quad ew_{1,g} = \left[\frac{\sum ew_{0,g'}}{\sum \sigma_{s,g'} ew_{0,g'}}\right] \sigma_{s,g} ew_{0,g}$$

Sum over all groups:

$$\sum ew_{1,g} = \left[\frac{\sum ew_{0,g'}}{\sum \sigma_{s,g'} ew_{0,g'}}\right] \sum \sigma_{s,g} ew_{0,g} = \sum ew_{0,g}$$

Method Confirmation (Testcase: *Cool*)



- Heating profile between either methods match very closely
- Also matching 15 group iron profile
- $T_{RAD,0} = 1.0$ keV
- $T_{MAT,0} = 0.01$ keV

Spectra Comparison (Testcase: *Cool*)

- Flocking: 10k particles
  Jayenne: 5 million particles
- 200 iron group material data
- $T_{RAD,0} = 1.0$ keV
- $T_{MAT,0} = 0.01$ keV
- Interesting spectra behavior due to being under being the smallest Planck Group